

УДК 519.6

ПРОГРАММА "ЭГИДА-ТЕСТ" ДЛЯ ЧИСЛЕННОГО ИССЛЕДОВАНИЯ ПАРАМЕТРОВ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В. Ю. Колобянин, И. Н. Чистякова
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Описывается программа ЭГИДА-ТЕСТ с двухуровневым распараллеливанием (MPI+OpenMP), разработанная для тестирования высокопроизводительных вычислительных систем, ориентированных на численное моделирование сложных физических процессов. Приводится описание газодинамических тестов, входящих в состав ЭГИДА-ТЕСТ, представлены функциональные возможности программы для анализа полученных результатов тестирования. Внимание уделено особенностям реализации системной части тестовой программы, которая позволяет рассматривать любой тест в программе ЭГИДА-ТЕСТ как набор мини-тестов с различными характеристиками.

Программа ЭГИДА-ТЕСТ является упрощенным аналогом методики моделирования течений многокомпонентной сплошной среды ЭГАК, что позволяет сделать прогноз эффективности работы реального приложения на исследуемой высокопроизводительной вычислительной системе.

Ключевые слова: высокопроизводительные вычислительные системы, методика ЭГАК, тестовая программа ЭГИДА-ТЕСТ, двухуровневое распараллеливание, типовая схема распараллеливания, оперативная память, система тестов, мониторинг тестирования, таймирование.

Введение

Для тестирования высокопроизводительных вычислительных систем (ВВС), ориентированных на численное моделирование сложных физических процессов, в РФЯЦ-ВНИИЭФ разработан пакет методических прикладных тестов [1]. Первые тесты были созданы в середине 1990-х годов. Для численного исследования параметров ВВС при решении задач газовой динамики использовалась тестовая программа ЭГИДА-ТЕСТ с одноуровневым распараллеливанием на распределенной памяти с интерфейсом передачи сообщений MPI.

Пакет методических прикладных тестов постоянно развивается, пополняется новыми тестами и адаптируется под новые аппаратно-программные платформы. Рассматриваемая в статье тестовая программа является новой версией программы ЭГИДА-ТЕСТ с двухуровневым распараллеливанием (MPI+OpenMP), написанной на языке C++.

В настоящее время тестовые программы широко используются на стадии выбора архитектуры перспективных ВВС, включая оценку производительности и эффективности микропроцессоров.

Учитывая, что программа ЭГИДА-ТЕСТ разработана на основе методики моделирования течений многокомпонентной сплошной среды ЭГАК [2], результаты тестирования даже небольшого экспериментального сегмента позволяют сделать прогноз эффективности работы реального приложения на ВВС, создаваемой на тестируемых микропроцессорах.

При создании новой программы ЭГИДА-ТЕСТ был учтен большой опыт тестирования ВВС с использованием предыдущей версии программы. В тестовую программу ЭГИДА-ТЕСТ добавлена возможность проверки правильности выполнения вычислений при проведении расчетов в различных режимах, разработан мониторинг тестирования, позволяющий получить обширную информацию для анализа тестируе-

мой системы и в случае неудовлетворительных результатов тестирования определить причины снижения ожидаемой эффективности.

В новой версии расширен набор тестов. При разработке нового теста "Движение сферической системы" была поставлена цель увеличить нагрузку на коммуникационную сеть и количество обращений к памяти. Кроме того, для всех тестов появилась возможность выполнять верификацию расчетов при различных режимах запуска теста (с варьированием количества MPI-процессов и OMP-потоков) на разных вычислительных системах.

В программе ЭГИДА-ТЕСТ моделируется трехмерный процесс газовой динамики, который, как и в предыдущей версии программы, состоит из двух этапов: лагранжева газовая динамика (лагранжев этап) и пересчет величин (эйлеров этап). Используется неподвижная счетная сетка, состоящая из кубов или параллелепипедов. Формулы, модели и предположения для процесса газовой динамики в ЭГИДА-ТЕСТ также остались прежними и описаны в [1]. В данной статье основное внимание уделено системной реализации программы ЭГИДА-ТЕСТ, ее функциональным возможностям и описанию газодинамических тестов, выбранных для тестирования.

В статье использованы термины и определения в соответствии с Государственным стандартом [3].

Особенности системной реализации программы ЭГИДА-ТЕСТ

Программа ЭГИДА-ТЕСТ является облегченной версией методики ЭГАК с минимальными функциональными возможностями. При этом системная реализация, типовая схема распараллеливания [4] и организация памяти у них идентичны. Расчеты по методике ЭГАК могут выполняться на адаптивно-встраиваемой дробной сетке [5] с различными уровнями дробления и динамической балансировкой вычислительной нагрузки [6]. Данная реализация потребовала сложной организации памяти, которая может оказывать значительное влияние на производительность теста в зависимости от архитектуры исследуемой ВВС. Для возможности прогнозирования эффективности работы методики ЭГАК на ВВС перспективной архитектуры необходимо было сохранить ту же организацию памяти в тестовой программе, что и в реальном прило-

жении, несмотря на отсутствие в ЭГИДА-ТЕСТ адаптивно-встраиваемой дробной сетки и балансировки.

Для объяснения ситуации кратко остановимся на организации памяти в методике ЭГАК. При запуске задачи программа сразу захватывает разрешенный в настройках к задаче объем памяти для пользовательского приложения и формирует свой пул памяти. Память, включая основной уровень ячеек, выделяется блоками по 8 ячеек ($2 \times 2 \times 2$). При дроблении материнской ячейки выделяется блок из восьми ячеек для ее дочерних ячеек, при удалении дочерних ячеек блок в пуле памяти помечается свободным. Балансировка работает с тем же пулом памяти, помечая блоки свободными при миграции ячеек на другие MPI-процессы или, наоборот, захватывая свободные блоки памяти при добавлении ячеек на текущем MPI-процессе. Данная технология позволяет экономично расходовать память, но при такой организации хранимая информация не может априори считаться упорядоченной по ячейкам (ячейки в памяти не размещены последовательно друг за другом), что на современных архитектурах может сильно снижать реальную производительность.

Но если заимствование организации памяти для тестовой программы можно в какой-то степени считать вынужденной мерой, то использование *типовой схемы распараллеливания* имеет большие преимущества при тестировании и анализе результатов тестирования.

Типовая схема распараллеливания применяется на каждом из этапов, полученных в результате разбиения всех счетных процессов согласно технологии *рамочного программирования*.

Стоит отметить, что основы рамочного программирования были разработаны еще для предшествующей версии методики ЭГАК с одноуровневой моделью распараллеливания на основе MPI [2, 6] и перенесены в тестовую программу ЭГИДА-ТЕСТ. При создании новой версии с двухуровневым распараллеливанием разработчикам удалось, существенно переработав системную часть, создать новую типовую схему [4] и, таким образом, сохранить практически неизменной счетную часть.

В соответствии с технологией рамочного программирования для каждого этапа программируется обработка одной точки¹ и указываются

¹Для процесса газовой динамики точкой является ячейка счетной области.

величины, которые необходимо передать на этом этапе в MPI-обменах. Вся остальная работа, включая двухуровневое распараллеливание, лежит на системной части и выполняется в типовой схеме распараллеливания. В многопоточном режиме все потоки на этом этапе последовательно выполняют функцию обработки одной точки из своего набора.

Таким образом, каждый этап является по сути отдельным тестом. А исходный тест можно рассматривать как набор различных тестов с разными характеристиками. Данная особенность была использована разработчиками при создании системы мониторинга тестирования.

Набор тестов в программе ЭГИДА-ТЕСТ

В новой версии программы ЭГИДА-ТЕСТ для тестирования используются два теста: "Седовский (точечный) взрыв" и "Движение сферической системы".

Выбор теста производится в зависимости от целей тестирования. Тесты могут запускаться как в режиме умножения (слабое масштабирование), когда задача масштабируется с увеличением количества вычислительных узлов ВВС, так и в режиме деления (сильное масштабирование), при котором размеры задачи остаются неизменными при переходе на другое количество вычислительных узлов.

При подборе оптимального режима запуска внутри узла (соотношение MPI-процессов и OMP-потоков на узле) размеры задачи не меняются.

Размер задачи и режим тестирования (умножения или деления) задаются в файле-инструкции счета задачи. Для удобства редактирования файл имеет текстовый формат. В режиме деления указывается количество ячеек задачи по каждому из трех пространственных направлений: N , M , K (общее количество ячеек $N \times M \times K$). В режиме умножения задается количество ячеек по каждому из трех направлений для одного MPI-процесса: n , m , k (общее количество ячеек $n \times m \times k$). Программа, используя размеры декартовой решетки MPI-процессов для текущего запуска задачи, автоматически формирует размеры счетной области задачи таким образом, чтобы на каждом MPI-процессе располагалось $n \times m \times k$ ячеек.

Результат текущего запуска теста фиксируется в выходном файле. В начале этого фай-

ла содержится информация по данному запуску: размер задачи, размер декартовой решетки MPI-процессов для текущего запуска задачи, режим запуска n MPI \times m OMP (количество MPI-процессов в запуске и количество потоков на каждом MPI-процессе), максимальный объем используемой оперативной памяти (ОП) для одного MPI-процесса, количество ОП на узле и количество свободной ОП на узле. Описание остальной справочной информации, содержащейся в выходном файле, для темы данной статьи не существенно и поэтому опускается.

Далее приведены постановки задач для двух указанных тестов и рассмотрены их преимущества и недостатки. Система единиц в тестах — безразмерная.

Тест 1. Седовский (точечный) взрыв.

Данная задача предложена и аналитически решена в работе [7]. Начальная геометрия задачи о точечном взрыве отражена на рис. 1, а.

Счетная область при моделировании точечного взрыва в газодинамическом приближении в общем случае представляет собой параллелепипед. Количество ячеек в каждом направлении может меняться при неизменном размере ячейки $1 \times 1 \times 1$. Таким образом, при количестве ячеек $N \times M \times K$ размеры счетной области: $0 < x < N$, $0 < y < M$, $0 < z < K$.

В области задано два вещества, для расчета используется уравнение состояния идеального газа $p = (\gamma - 1) \rho e$, $\gamma = 1,4$.

В подобласти $x < 1$, $y < 1$, $z < 1$ задается вещество 1 с плотностью $\rho = 1$, энергией $e = 1$, скоростью $U = 0$. Остальная область заполнена веществом 2 с плотностью $\rho = 1$, энергией $e = 0$, скоростью $U = 0$. На всех границах области задано граничное условие *жесткая стенка*.

В результате действия начального энерговыделения происходит формирование ударной волны, которая распространяется в пространстве практически сферически-симметрично за счет малости начальной области энерговыделения (рис. 1, б).

Достоинством данного теста является его хорошая масштабируемость при сбалансированной вычислительной нагрузке. Таким образом, в режиме слабого масштабирования этот тест легко запускается на различном количестве вычислительных узлов исследуемой ВВС.

Тест 1 в режиме слабого масштабирования хорошо использовать для тестирования коммуни-

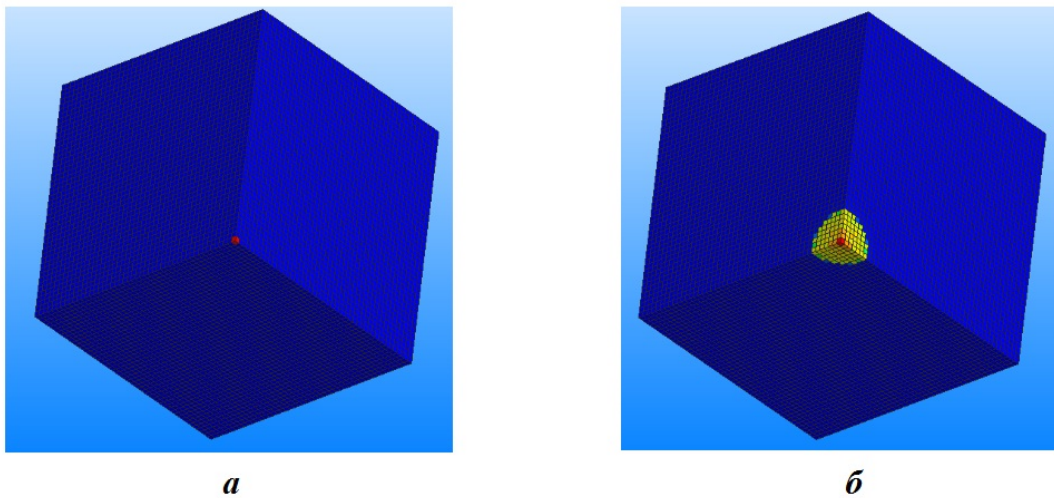


Рис. 1. Тест 1. Средняя энергия: $a - t = 0$; $b - t = 100$

кационной сети ВВС, так как при увеличении количества вычислительных узлов для запуска теста нагрузка на коммуникационную сеть возрастает.

Недостатком этого теста является небольшое количество веществ (всего два) в задаче. Вследствие этого на нем могут не обнаружиться проблемы, возникающие в реальных задачах с большим количеством веществ.

Руководствуясь этим, в новой версии программы ЭГИДА-ТЕСТ в дополнение к данному тесту был разработан тест "Движение сферической системы".

Тест 2. Движение сферической системы.

Размеры счетной области: $-17,5 < x < 17,5$; $0 < y < 17,5$; $0 < z < 17,5$.

Всего в задаче задано 13 веществ (табл. 1), для всех веществ задано уравнение состояния идеального газа с $\gamma = 1,4$. Система из 12 сферических слоев (центр в точке $(0, 0, 0)$, толщина каждого слоя равна 1) помещена в область, занятую воздухом (вещество 13). Во всех ячейках линейной интерполяцией задается энергия $E(r)$. В результате в веществах задачи реализуются давления, приводящие к движению слоев системы. Граничные условия: $E(r = 0) = 0$; $E(r = 30) =$

$= 10$, где $r = 30$ обеспечивает охват всей счетной области. На всех границах области задано граничное условие *жесткая стенка*.

Постановка задачи для теста 2 отражена на рис. 2.

Недостаток теста 2 в том, что он не так хорошо масштабируется в режиме умножения, как тест 1.

Во-первых, в тесте 2 с увеличением количества ячеек уменьшаются размеры ячейки при постоянных размерах счетной области задачи.

Во-вторых, с увеличением количества компонентов в задаче увеличивается дисбаланс вычислительной нагрузки. Но за счет того, что для всех веществ задано одно и то же уравнение состояния идеального газа, дисбаланс вычислительной нагрузки существенно уменьшен.

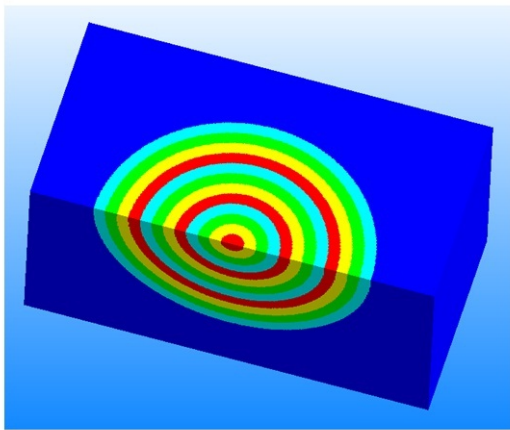
В-третьих, и это самый большой недостаток при масштабировании теста 2 в режиме умножения, автоматическое масштабирование для него не поддерживается. Для этого теста в файле-инструкции к запуску необходимо для каждой задачи задавать количество ячеек по каждому направлению.

Рассмотрим три варианта слабого масштабирования этого теста с учетом того, что соотно-

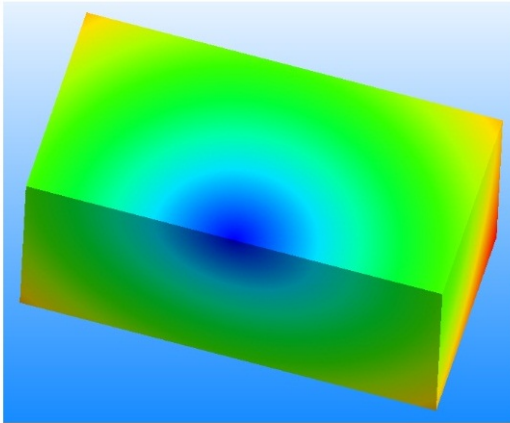
Таблица 1

Тест 2. Начальная постановка задачи

Номер вещества	1	2	3	4	5	6	7	8	9	10	11	12	13
Плотность вещества	20	3	2	1	20	3	2	1	20	3	2	1	0,001



а



б

Рис. 2. Тест 2. Постановка задачи: *а* — концентрация веществ; *б* — средняя энергия

шение геометрических размеров счетной области 2:1:1 не должно меняться.

Вариант 1. При масштабировании количество ячеек на каждом MPI-процессе равно $n \times n \times n$, т. е. по всем направлениям одинаковое.

В этом случае соотношение 2:1:1 необходимо соблюдать для декартовой решетки MPI-процессов, на которой будет выполняться расчет.

Кубическая счетная область $n \times n \times n$ на MPI-процессе задана исходя из того, что это оптимальный вариант для MPI-передач. Но, учитывая, что общее количество MPI-процессов для задачи кратно количеству MPI-процессов на узле, на указанные выше ограничения еще будут накладываться ограничения в выборе режима запуска на узле (количество MPI-процессов и OMP-потоков на узле).

Рассмотрим это на примере тестирования узла, имеющего 128 ядер. Выбранные режимы за-

пуска: 16 MPI×8 OMP и 8 MPI×16 OMP на узле. Для этого примера в табл. 2 приведены все варианты запуска теста 2 в режиме слабого масштабирования с учетом сохранения соотношения 2:1:1 для декартовой решетки MPI-процессов. Как видно из табл. 2, часть вариантов отпадает, так как число MPI-процессов в задаче не кратно количеству MPI-процессов на узле.

Вариант 2. Количество ячеек на MPI-процессе равно $2n \times n \times n$. В этом случае декартова решетка MPI-процессов по всем трем направлениям должна быть одинаковой. Например, $1 \times 1 \times 1$, $2 \times 2 \times 2$, $4 \times 4 \times 4$ и т. д.

Проблема при использовании этого варианта, как и варианта 1, в том, что невозможно выполнить запуск теста 2 на произвольно заданном количестве вычислительных узлов.

Вариант 3. Общее количество ячеек в задаче увеличивается в том же соотношении, что и количество вычислительных узлов. Соотношение геометрических размеров счетной области 2:1:1 не должно меняться. Таким образом, запуск можно выполнять на произвольно заданном количестве вычислительных узлов. Этот вариант является самым грубым, так как количество счетных ячеек на MPI-процессах может быть различным в одном запуске и могут меняться размеры MPI-сообщений между соседними MPI-процессами по сравнению с предыдущими запусками.

Таблица 2

Варианты запуска теста 2 в режиме слабого масштабирования при количестве ячеек на MPI-процессе, равном $n \times n \times n$

Решетка MPI- процессов	Кол-во MPI	Кол-во узлов	
		Режим 16 MPI×8 OMP	Режим 8 MPI×16 OMP
$2 \times 1 \times 1$	2	1/8	1/4
$4 \times 2 \times 2$	16	1	2
$6 \times 3 \times 3$	54	—	7
$8 \times 4 \times 4$	128	8	16
$10 \times 5 \times 5$	250	—	—
$12 \times 6 \times 6$	432	27	54
$14 \times 7 \times 7$	686	—	—
$16 \times 8 \times 8$	1024	64	128
$18 \times 9 \times 9$	1458	—	—
$20 \times 10 \times 10$	2000	125	250

На кластерах с небольшим количеством вычислительных узлов тест 2 можно запускать в режиме деления.

При сравнительном анализе узлов с микропроцессорами разного типа использование теста 2 является более предпочтительным, чем теста 1, а проблемы с масштабированием теста 2 исчезают, так как размеры задачи не меняются.

Мониторинг выполнения тестов в программе ЭГИДА-ТЕСТ

При анализе вычислительных узлов с различными типами микропроцессоров часто недостаточно получить коэффициенты ускорения или замедления работы теста на исследуемых микропроцессорах. Желательно иметь дополнительную информацию, чтобы понять, что повлияло на полученный результат.

Для этого при тестировании запускают набор тестов с различными характеристиками, например по интенсивности вычислений, объему MPI-трафика, количеству обращений к памяти.

Уникальность системной части программы ЭГИДА-ТЕСТ в том, что благодаря рамочному программированию любой счетный процесс на стадии разработки алгоритма разбивается на отдельные блоки. Каждый блок — это отдельный этап, обрабатываемый типовой схемой распараллеливания. Например, процесс газовой динамики разбит на несколько десятков этапов. Каждый этап при запуске теста 1 или теста 2 можно считать отдельным тестом со своими характеристиками, основные из которых — интенсивность вычислений, процент условных операций, доля векторных инструкций, объем считанной и записанной информации в память и объем пересылаемых MPI-сообщений. Эти параметры, характеризующие каждый этап, были получены при профилировании тестов 1 и 2. Заметим, что для получения объема MPI-трафика профилирование делать не обязательно: количество передаваемых величин для одной ячейки указывается как параметр этапа для типовой схемы распараллеливания.

Имея возможность при замерах для каждого этапа фиксировать полное время его выполнения, время счета, время, затраченное на MPI-обмены и т. д., можно получить коэффициенты ускорения/замедления по каждому этапу при сравнительном анализе микропроцессоров разных типов. Знание характеристик каждого эта-

па дает возможность проанализировать причину ускорения или замедления программы ЭГИДА-ТЕСТ на исследуемом микропроцессоре.

Суть мониторинга состоит в том, чтобы по каждому этапу фиксировать и выдавать минимальное/максимальное время счета, номер и тип MPI-процесса², на котором получено такое время. Аналогичная информация собирается по MPI-обменам и накладным расходам по упаковке/распаковке буферов MPI-сообщений.

При этом, учитывая мелкозернистое поточечное распараллеливание, заложенное в типовую схему распараллеливания, важно получить корректные засечки времени. При засечках времени таймер может многократно переходить из активного состояния в пассивное и наоборот, и неправильное определение места таймирования даже при использовании оптимального таймера может привести к неправильным результатам таймирования. Кроме того, разрешение таймера должно быть достаточным для точного таймирования небольших участков кода, а накладные расходы при использовании таймера должны быть незначительными.

Этому вопросу уделялось большое внимание при разработке старой версии программы ЭГИДА-ТЕСТ, все наработки перешли в системную часть новой версии программы, но были расширены и усовершенствованы.

Для уменьшения накладных расходов под операционной системой Linux/Unix используется наносекундный таймер, который при переходе из активного состояния в пассивное фиксирует только количество тиков в активном состоянии, а перевод тиков в секунды выполняется только при запросе `Answer`.

Таймирование в программе ЭГИДА-ТЕСТ, так же, как и в методике ЭГАК, выполняется всегда и составляет тысячные доли процента от времени выполнения теста.

Ограничений на количество таймеров нет (новый таймер — отдельный объект класса `TimingClass`), что очень удобно при исследовательской работе.

Режим мониторинга задается в настройках условной компиляции программы ЭГИДА-ТЕСТ. Мониторинг может задаваться выборочно для определенного диапазона счетных шагов.

²Тип MPI-процесса важен для машин с гибридной архитектурой. Так, тип *U* означает, что MPI-процесс выполняется на универсальном процессоре, *C* — на сопроцессоре.

При этом необходимо учитывать, что время счета шагов, для которых задан мониторинг, увеличивается. Это связано, во-первых, с тем, что для получения достоверной информации по каждому этапу команда `MPI_Barrier` стоит в конце каждого вычислительного этапа. Во-вторых, в режиме мониторинга каждый этап завершается коллективной операцией по сбору минимального/максимального времени выполнения этапа с определением номера и типа MPI-процесса и дальнейшим выводом этой информации в файл `root`-процессом.

В табл. 3 показана часть детальной информации, предоставляемой в результате мониторинга времени счета по отдельным вычислительным этапам процесса газовой динамики на гибридных узлах с универсальными процессорами и сопроцессорами. Мониторинг выполнялся на тесте 2. Кроме минимального и максимального времени выполнения этапа (t_{\min} , t_{\max}), а также номера MPI-процесса, на котором были зафиксированы эти времена, и его типа, приводится информация о номере вычислительного этапа, его имени и количестве передаваемых значений в ячейке (N_{send}) для гранично-процессорных ячеек. Если значение нулевое, передачи отсутствуют.

ЭГИДА-ТЕСТ также используется как испытательный полигон для оптимизации методики ЭГАК и адаптации ее к ВВС перспективной архитектуры. Например, зная коэффициент ускорения от использования гибридного поля по каждому этапу на тестируемой ВВС с гибридной

архитектурой и имея информацию о типе процессора, на котором было зафиксировано t_{\max} , можно понять, насколько правильно сбалансирована вычислительная нагрузка внутри узла и между узлами, а также какие этапы необходимо оптимизировать, чтобы увеличить производительность счета по методике ЭГАК на ВВС перспективной архитектуры.

Такой же подход применяется, когда необходимо определить, на каких этапах при тестировании снижается производительность счета. В этом случае анализируются коэффициенты ускорения тестируемой ВВС относительно существующих. Часто нарушение равновесия между пиковой производительностью микропроцессоров, пропускной способностью памяти и коммуникационной сети приводит к существенному снижению ожидаемого ускорения на новой ВВС. Исходя из основных характеристик этапа, который показывает низкое ускорение, соответствующий блок кода оптимизируется либо разрабатывается новый алгоритм, который будет лучше учитывать особенности ВВС перспективной архитектуры.

Заключение

Создание новой версии тестовой программы ЭГИДА-ТЕСТ с двухуровневым распараллеливанием обеспечило возможность исследования перспективных ВВС гибридной архитектуры (в

Таблица 3

Тест 2. Мониторинг времени счета на примере отдельных этапов газовой динамики

Номер этапа	t_{\min}	MPI-процесс		t_{\max}	MPI-процесс		Название этапа	N_{send}
		номер	тип		номер	тип		
...								
15	0,073	28	C	0,09	42	U	SetCoorVolMainLevel	0
16	0,002	28	C	0,004	137	U	Preparation_Etap_Transport_0	0
17	0,045	28	C	0,053	109	C	Etap_Transport_0	0
18	0,092	28	C	0,135	73	U	Preparation_Etap_Transport_1	0
19	0,25	28	C	0,317	105	U	Etap_Transport_1	31
23	0,047	28	C	0,078	233	U	Preparation_Etap_Transport_2	0
24	0,103	28	C	0,151	105	U	Etap_Transport_2	14
25	0,011	100	C	0,033	218	U	Preparation_Etap_Transport_3	0
26	0,07	0	U	0,158	132	C	Etap_Transport_3	0
27	0,003	224	U	0,004	109	C	Preparation_Etap_Transport_4	0
28	0,149	224	U	0,196	71	C	Etap_Transport_4	25
...								

том числе с использованием компонентов отечественной разработки), а новые функциональные возможности программы и разработанная система мониторинга расширили сферу ее применения. На стадии выбора архитектуры перспективных ВВС программа ЭГИДА-ТЕСТ может использоваться для поиска оптимальных аппаратно-программных решений, а также для адаптации методики ЭГАК к выбранной архитектуре с целью повышения ее эффективности.

Исследование выполнено в рамках научной программы Национального центра физики и математики (проект "Национальный центр исследования архитектур суперкомпьютеров").

Список литературы

1. Алексеев А. В., Беляев С. П., Бочков А. И., Быков А. Н., Ветчинников М. В., Залылов А. Н., Нуждин А. А., Огнев С. П., Самсонова Н. С., Сапронов Н. С., Чистякова И. Н., Шемякина Т. В., Шагаллиев Р. М., Янилкин Ю. В. Методические прикладные тесты РФЯЦ-ВНИИЭФ для численного исследования параметров высокопроизводительных вычислительных систем // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 2. С. 86–100.
Alekseev A. V., Belyaev S. P., Bochkov A. I., Bykov A. N., Vetchinnikov M. V., Zalyalov A. N., Nuzhdin A. A., Ognev S. P., Samsonova N. S., Sapronov N. S., Chistyakova I. N., Shemyakina T. V., Shagaliev R. M., Yanilkin Yu. V. Metodicheskie prikladnye testy RFYaTs-VNIIEF dlya chislenogo issledovaniya parametrov vysokoproizvoditelnykh vychislitelnykh system // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2020. Vyp. 2. S. 86–100.
2. Янилкин Ю. В., Беляев С. П., Бондаренко Ю. А., Гаврилова Е. С., Гончаров Е. А., Горбенко А. Д., Городничев А. В., Губков Е. В., Гузнова А. Р., Дегтяренко Л. И., Жарова Г. В., Софронов В. Н., Стадник А. Л., Ховрин Н. А., Чернышова О. Н., Чистякова И. Н., Шемяков В. Н. Эйлеровы численные методики ЭГАК и ТРЭК для моделирования многомерных течений многокомпонентной среды // Труды РФЯЦ-ВНИИЭФ. 2008. Вып. 12. С. 54–65.

Yanilkin Yu. V., Belyaev S. P., Bondarenko Yu. A., Gavrilova E. S., Goncharov E. A., Gorbenko A. D., Gorodnichev A. V., Gubkov E. V., Guzhova A. R., Degtyarenko L. I., Zharova G. V., Sofronov V. N., Stadnik A. L., Khovrin N. A., Chernyshova O. N., Chistyakova I. N., Shemyakov V. N. Eylerovy chislennyye metodiki EGAK i TREK dlya modelirovaniya mnogomernykh techeniy mnogokomponentnoy sredy // Trudy RFYaTs-VNIIEF. 2008. Vyp. 12. S. 54–65.

3. Высокопроизводительные вычислительные системы. Термины и определения. ГОСТ Р 57700.27-2020. М.: Стандартинформ, 2020.
Vysokoproizvoditelnye vychislitelnye systemy. Terminy i opredeleniya. GOST R 57700.27-2020. M.: Standartinform, 2020.
4. Колобянин В. Ю., Федоров А. А., Антипина Н. Р. Двухуровневое распараллеливание явных разностных схем методики ЭГАК // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2017. Вып. 3. С. 62–69.
Kolobyandin V. Yu., Fyedorov A. A., Antipina N. R. Dvukhurovnevoe rasparallelivaniye yavnykh raznostnykh skhem metodiki EFAK // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2017. Vyp. 3. S. 62–69.
5. Янилкин Ю. В., Беляев С. П., Городничев А. В., Волгин А. В., Воронов Е. Г., Гузнова А. Р., Дегтяренко Л. И., Жарова Г. В., Кучерова П. А., Стадник А. Л., Ховрин Н. А. Комплекс программ ЭГАК++ для моделирования на адаптивно-встраиваемой дробной счетной сетке // Там же. 2003. Вып. 1. С. 20–28.
Yanilkin Yu. V., Belyaev S. P., Gorodnichev A. V., Volgin A. V., Voronov E. G., Guzhova A. R., Degtyarenko L. I., Zharova G. V., Kucherova P. A., Stadnik A. L., Khovrin N. A. Kompleks programm EGAK dlya modelirovaniya na adaptivno-vstraivayushcheysya drobnou schyetnoy setke // Tam zhe. 2003. Vyp. 1. S. 20–28.
6. Беляев С. П. Метод мелкозернистого распараллеливания с динамической балансировкой на примере задачи газовой динамики и вычислительные эксперименты на параллельной системе // Там же. 2000. Вып. 1. С. 45–49.

Belyaev S. P. Metod melkozernistogo raspallelivaniya s dinamicheskoy balansirovkoy na primere zadachi gazovoy dinamiki i vychislitelnye eksperimenty na parallelnoy sisteme // Tam zhe. 2000. Vyp. 1. S. 45–49.

в механике. М.: Наука, 1965.

Sedov L. I. Metody podobiya i razmernosti v mekhanike. М.: Nauka, 1965.

7. *Седов Л. И.* Методы подобия и размерности

Статья поступила в редакцию 22.08.22.
